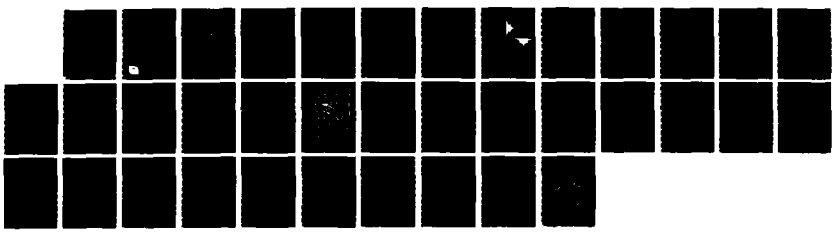
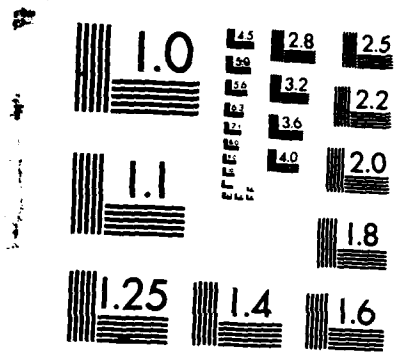
AD-A173 928 STARS (SOFTWARE TECHNOLOGY FOR ADAPTABLE RELIABLE SYSTEMS) GOALS AND OBJE (U) INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA P FONASH ET AL SEP 85 1/1

UNCLASSIFIED IDA-P-1855 IDA/HQ-85-30153 MDA903-84-C-0031 F/G 9/2 NL





AD-A173 928

IDA PAPER P-1855

STARS GOALS AND OBJECTIVES WORKING GROUP: FINAL REPORT

Peter Fonash
Chuck Lillie
John Machado
Samuel T. Redwine, Jr.
William Riddle
Dennis Turner

September 1985

Prepared for
Office of the Under Secretary of Defense for Research and Engineering

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

IDA Log No. HQ 85-30153

SECURITY CLASSIFICATION OF THIS PAGE AD-A173928

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) P-1855	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Institute for Defense Analyses 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION DoD-IDA Management Office (OUSDRE)
6c. ADDRESS (City, State, and ZIP Code) 1801 N. Beauregard Street Alexandria, VA 22311	7b. ADDRESS (City, State, and ZIP Code) 1801 N. Beauregard Street Alexandria, VA 22311
8a. NAME OF FUNDING/SPONSORING ORGANIZATION (R&T) STARS Joint Program Office 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA 903 84 C 0031
8c. ADDRESS (City, State, and ZIP Code) 1211 Fern Street Arlington, VA 22202	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. T-4-236 / WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) STARS Goals and Objectives Working Group Final Report (U)	
12. PERSONAL AUTHOR(S) Peter Fonash, Chuck Lillie, John Machado, Samuel T. Redwine, Jr., William Riddle, Dennis Turner	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM ____ TO ____	14. DATE OF REPORT (Year, Month, Day) September 1985 15. PAGE COUNT 40
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB-GROUP	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Goal Programming; Software; STARS; Management Planning and Control; Software Technology

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

At the direction of the Acting Director of the Department of Defense (DoD) Software Technology for Adaptable Reliable Systems (STARS) Program, a Goals and Objectives Working Group was formed to prepare this decomposition of the program's goals and objectives. Starting with the STARS Program's Charter, the Group developed statements of derivative, high-level, technical goals and objectives for the Program as a whole, and decomposed these goals and objectives into specific statements of the technical scope and goals of each Activity Area within the STARS Program. This document also addresses issues such as phasing of "products," coordination with other programs, and demonstration of cost/benefit.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) / 22c. OFFICE SYMBOL

IDA PAPER P-1855

STARS GOALS AND OBJECTIVES WORKING GROUP: FINAL REPORT

Peter Fonash
Chuck Lillie
John Machado
Samuel T. Redwine, Jr.
William Riddle
Dennis Turner

September 1985

IDA
INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 84 C 0031
Task T-4-236

Table of Contents

Figures

1. Introduction

In February 1985, Joseph Batz, Acting Director of the Department of Defense (DoD) Software Technology for Adaptable, Reliable Systems (STARS) Program, formed a Goals and Objectives Working Group and asked it to prepare a decomposition of the Program's goals and objectives. The task was to start with the Charter, develop statements of derivative, high-level, technical goals and objectives for the Program as a whole, and then decompose these goals and objectives into specific statements of the technical scope and goals of each Activity Area within the STARS Program. In addition, the Group was asked to address issues such as phasing of "products," coordination with other programs, and demonstration of cost/benefit.

The Goals and Objectives Working Group held three meetings, one each in the months of February, March and April. In the course of these meetings, the Group evolved statements for the Program's high-level, technical goals and objectives and for the technical scope and goals of the individual Activity Areas. These statements and the logic of their derivation from the STARS Program's Mission Statement are the subject of this final report.

The main sections of this final report, relating the derivation of Activity Area technical goals from the STARS mission statement, reflect a consensus within the Goals and Objectives Working Group. The Group was unable to quantify these goals, either for the individual Activity Areas or the Program as a whole. Quantification was discussed but the results were a reaffirmation of the near-impossibility of meaningful, defensible quantification at this point in time and a feeling that quantification of the Program's and Areas' goals must be a part of the program itself. This quantification concomitant with the first stages of the Program is discussed at appropriate spots in this final report.

During the Group's discussions, several topics arose for which no definitive conclusions were reached. These topics pertain to the managerial and coordination issues mentioned above. The Group's thinking on these topics is related in the final section of this report.

2. STARS Program's Charter

The STARS Program's mission and goals are specified in the chartering document signed by Undersecretary of Defense for Research and Engineering DeLauer on 1 November 1984. To establish a context for this report's discussion of technical goals and objectives, we quote the pertinent section of this chartering document:

> "Future defense systems will utilize more software of higher complexity and will require greater reliability

and adaptability of such software than is presently achieved. Advances in computer software and system technology now offer opportunities for new and enhanced defense capabilities involving satellite, missile, shipboard, aircraft, battlefield command and control, intelligence and other defense areas. All of these will require the development of significant quantities of software for the related mission critical computer systems.

The DoD Software Initiative (STARS) has been established to develop the technology which is required and to more rapidly transition such technology into use. STARS will improve the process of software engineering to increase the adaptability and reliability of mission critical software. STARS will develop and manage programs to achieve dramatic improvement in our ability to provide software meeting defense mission requirements. The STARS goals are: (1) improve productivity, (2) improve quality and reliability, (3) promote the development and application of reusable software, and (4) reduce the time and cost of developing defense software. ..."

3. STARS Program's Mission

The STARS Program's mission is specified in the chartering document. A discussion of the mission is included here to set the stage for the remainder of this report. Ample rationale for the Program and its mission may be found in other documents.

Briefly, the STARS Program's mission is:

improve the DoD's ability to provide adequate, effective and upgradable defense system software.

(See Figure 1.) "Adequate" means the software meets the increasingly demanding requirements stemming from an increased reliance on software to deliver defense system functionality. "Effective" means the software unfailingly meets its requirements throughout its full operational life span from initial deployment to final retirement from service. "Upgradable" means both major and minor changes in requirements can be expeditiously met by modifying existing software.

The ultimate concern is that defense systems, themselves, exhibit desirable characteristics, such as reliability. But, while there is this concern for the overall system, the focus is upon the software that, to an ever increasing degree, provides the system's functionality and determines its other, non-functional characteristics.

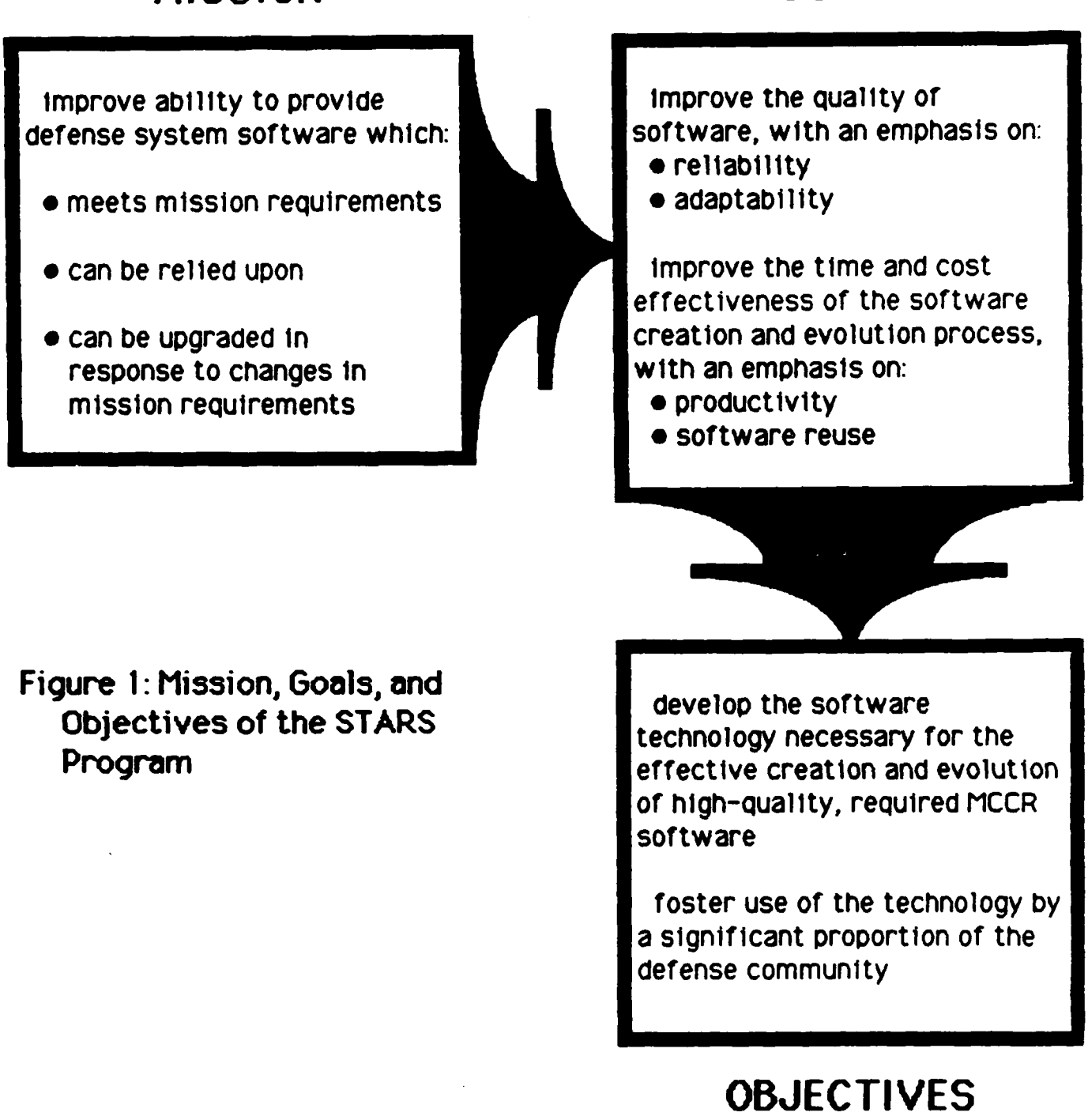


Figure 1: Mission, Goals, and Objectives of the STARS Program

4. STARS Program's Technical Goal

The DeLauer chartering document also singles out several specific, technical issues to be given direct attention. These issues, in essence, define the Program's high-level, technical goal. The issues, their relationship to the Program's mission, and the resulting goal are indicated in Figure 1 and discussed in this section. Again, this discussion is included for completeness and more detailed discussion and rationale may be found in other documents.

In general terms, the STARS Program's goal is to significantly improve the software creation and evolution process and the products resulting from it. Given the Program's mission, the products are an obvious focus of attention. The process itself is of interest because providing effective and upgradable software demands that attention be given to the process by which software matures over its full life span.

History has shown that the time and cost effectiveness of the process and the quality of the resulting products are particular problem areas. These process and product attributes therefore receive special attention within the STARS Program.

Time and cost effectiveness are particularly sensitive to the productivity of the people performing the process. They are also particularly sensitive to the extent to which parts of existing software systems may be reused in the development of new software systems. Productivity and reusability are emphasized in the STARS Program because of the large contribution their improvement can make to the process' overall time and cost effectiveness.

Effectiveness and upgradability are general, intuitive attributes and must be expressed in more specific terms to be effectively addressed. A software system's reliability is particularly relevant to its effectiveness. Without a high degree of reliability, a software system is unlikely to be judged effective no matter how well it meets its functional and performance requirements. And the system's adaptability is particularly relevant to its upgradability. Without a high degree of adaptability, it will be extremely difficult to upgrade a software system in response to changes in its requirements. Software reliability and adaptability therefore receive special attention within the STARS Program.

The technical goal of the STARS Program as a whole is:

> significantly improve the productivity and reusability attributes of the creation and evolution process for defense systems software and the reliability and adaptability attributes of the process' products.

5. STARS Program's Objectives

Improved productivity, reusability, reliability and adaptability cannot be sought in isolation. Practices designed to increase software reliability may, for example, negatively impact productivity. Thus, improvement in both the process and the products requires a coherent attack on all four attributes in tandem. In this section, we argue that this coherent attack can be accomplished by focusing on automation-oriented technology and state the objectives of the STARS Program in terms of this approach.

5.1. Value of a Technology-based Approach

A particularly effective way to simultaneously address a variety of process and product attributes is to focus on the technology supporting software creation and evolution. The technology deals with techniques and procedures which are relatively well-defined compared to the attributes of the products and processes resulting from the technology's use. Problems and solutions are more readily defined and assessed in the domain of technology; and it is easier to address tradeoffs and compatibility issues. Even with the extra effort needed to demonstrate that the technology leads to process and product improvement, the overall effort can turn out to be less than if one attempts to grapple with the problems, solutions, trade-offs and compatibility issues directly in the domain of the process and product attributes themselves. Thus, working in the technology domain can lead to to more timely results. Often, working in this domain provides results when they cannot be obtained by working in the attribute domain directly.

By adopting a technology-based approach, the objectives of the STARS Program can be expressed in terms of software creation and evolution technology rather than in terms of the process and product attributes targeted for improvement by the Program's technical goal. As a side-effect, the objectives must include the task of demonstrating a beneficial effect on the targeted process and product attributes.

5.2. General Objectives

The Program's general objectives, reflecting its goals, are given in Figure 1. One general objective is:

> develop technology leading to time and cost effective creation and evolution of high-quality defense system software.

Because of the emphasis on productivity, reusability, reliability and adaptability, technology pertaining to these attributes is of particular interest. And because developing new and improved technology is meaningless unless the technology is used, a second, but equally important, general objective is:

achieve widespread use of the technology throughout the defense community.

5.3. Value of Focusing on Automation-oriented Technology

Meeting these general objectives in the relatively short time frame of the STARS Program requires pursuing an approach promising a high return on investment in a short period of time. Experience has shown automation can lead to significant increases in the time and cost effectiveness of creation and evolution processes for artifacts such as cars, generally with a concomitant improvement in the general quality of the artifacts. Experience has also shown that, while the manufacturing portions of the process are most amenable to automation, it is possible to provide valuable assistance for other, creative activities; computer-aided design systems are an example. Because of the success, for software artifacts, of automated assistance such as provided by compilers, it is reasonable to expect a significant return on investment by pursuing an automation-oriented approach to meeting the STARS objectives. Because this will tend to rather rapidly reduce the currently high level of labor intensiveness of software creation and evolution, it is also reasonable to expect significant results in a short period of time.

Because the STARS Program's goal emphasizes productivity and reusability, the Program's objectives should emphasize automated assistance contributing to improving these process attributes. Because the STARS Program's goal also emphasizes reliability and adaptability, the search for automated assistance can be further focused upon those parts of the process most pertinent to these product attributes.

5.4. Meeting the Need to Achieve Wide-spread Use

Because the STARS Program must achieve wide-spread use of the technology it develops, the automated assistance must be attractive, rather than merely effective, so it can be more easily transitioned into wide-spread use. Studies of technology transition have consistently noted that introduction of new technology depends as much, and in some cases more, on its usability and level of productization than it does on the benefit the new technology provides. In fact, failure to consider the human factors of new technology in general, and automated assistance in particular, can so negatively impact its acceptability that widespread use is virtually impossible to achieve.

The breadth of use throughout the work force is critically dependent on a number of other factors. Primary among them is the ability of the members of the work force to use the automated assistance. This is a particular concern when the assistance embodies leading edge technology, for example, when automated

tools are provided for formally validating software performance attributes.

Another factor is the degree to which the community is motivated to use the automated assistance. In situations where there is a "purchaser" and a "supplier" of software, motivation to use demonstrably effective automated assistance, leading to cost and time-to-delivery advantages, can be provided by instituting various incentives and policies. In the Government arena, there is already a large body of policies, and regulations which implement them, protecting the interests of the Government as a "purchaser." Few of these have been defined with consideration of the particular problems of software acquisition in mind. Hence, there is the problem of improving existing policies in addition to instituting new ones.

A third factor affecting breadth of use is the degree to which the insertion process itself is actively "managed." This management can be relatively passive, involving, for example, the development of cost/benefit demonstrations or the provision of maintenance support for the software delivering the automated assistance. Alternatively, the management can be quite active, involving the promotion and marketing of the assistance.

5.5. Additional Work Force-related Concerns

To meet the STARS Program's goal, two additional work force-related concerns must be addressed. First, while automated assistance will improve the overall productivity of the work force, it will not necessarily lead to an ability to meet the demand. Therefore, it may be necessary to simultaneously take steps to increase the work force's size.

Second, it may be necessary to modify the work force's composition so there are adequate numbers of personnel for each of the various tasks involved. Consideration of the work force's composition is particularly important if new technology results in fundamental changes in the way software is created and evolved.

5.6. Specific Objectives

The general objectives of the STARS Program -- to develop the necessary technology and cause it to be widely used throughout the community -- translate to more specific objectives as indicated in Figure 2. These more specific STARS Program objectives are:

develop attractive assistance to software creation and evolution capitalizing on the benefit to be realized from automation,

improve the size and composition of the work force and the general ability to use the automated assistance,

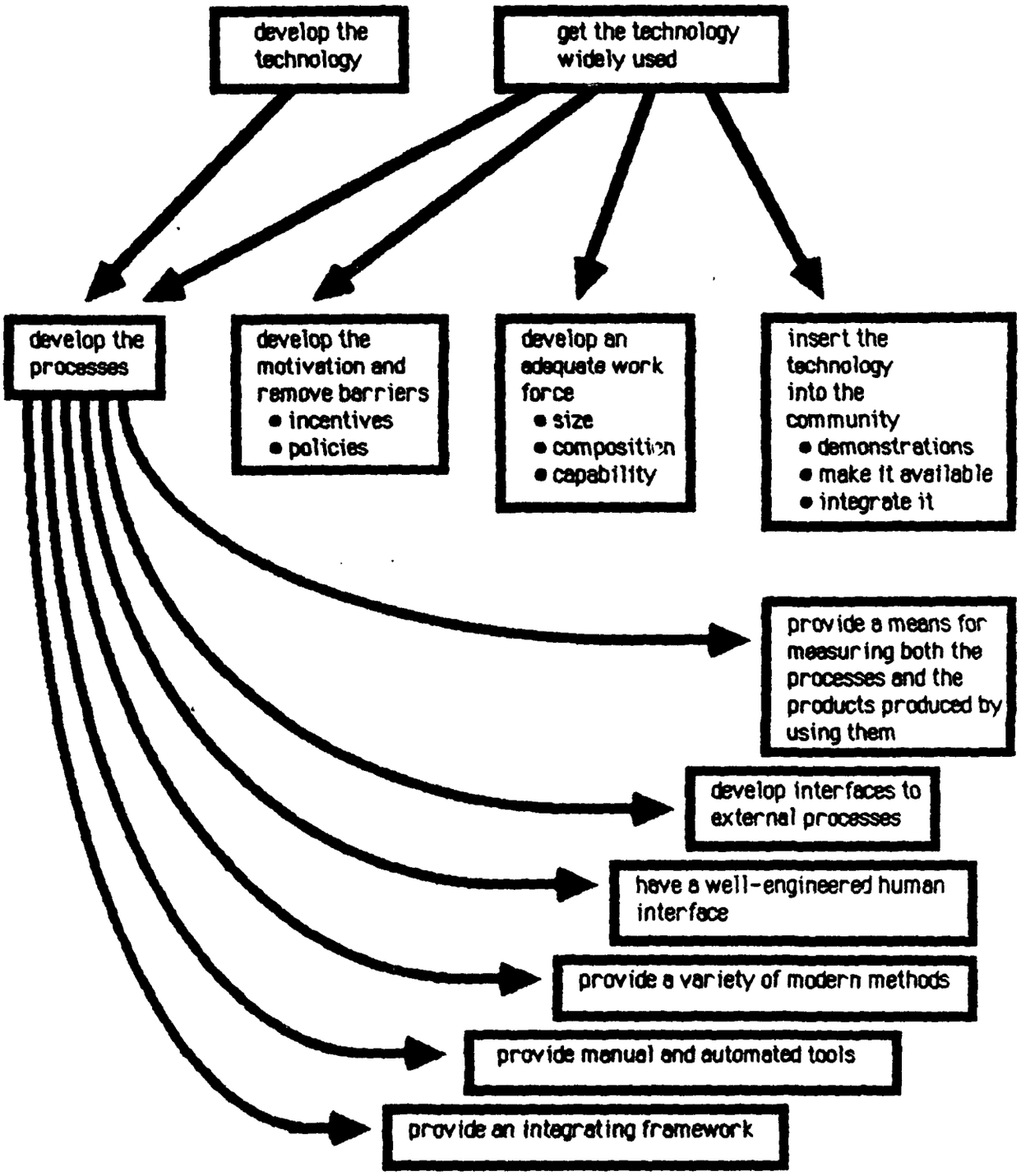


Figure 2: STARS Technical Objectives

refine existing "purchaser" incentives and policies, and define new ones, so "suppliers" are motivated to use the automated assistance, and

foster use of the automated assistance by actively managing its insertion into the community.

5.7. Refinement of the First Specific Objective

The first of these specific STARS Program objectives calls for identifying or developing automated assistance capitalizing on the benefit to be realized from automation. In addition, it calls for this automated assistance to be attractive so it will be widely accepted and used throughout the community. In Figure 2, this objective is further decomposed into a number of more detailed objectives, discussed in this section.

5.7.1. Focus on Tools

This decomposition requires consideration of just how the automated assistance can be most effectively made available for use. In the world of software, "automation" implies the preparation of computer programs embodying some procedure. For the support of software creation and evolution, these programs, typically called "automated tools," embody a procedure for performing some pertinent task, for example, the analysis of the flow of data through a piece of software. Most typically, it is impossible to completely automate all procedures or even all of a specific procedure. Additional, complementary, manual tools are, therefore, needed to make the automated tools effectively usable for the tasks occurring during software creation and evolution. In the remainder of this report, the term "tools" should be taken to mean both manual and automated tools.

A wide variety of activities occur during the life span of a typical piece of software and these activities are performed by an equally wide variety of personnel. The activities include: definition of the software's requirements, preparation of an architectural design giving the software's general structure and logic, detailed design of the data structures manipulated by the software and the algorithms used in this manipulation, preparation of the software's code, periodic validation of decisions, project team management, allocation of project resources, and investigation of various options. The personnel include: analysts, designers, programmers, project managers, acquisition managers, validators, trainers, and users. To be truly effective, tools must span these activities as broadly as possible and meet the needs of as many of the personnel as possible.

In striving for broad coverage of activities and personnel, there is a risk the individual tools will be incompatible in the

sense that information prepared by using one tool cannot be utilized by other tools. Even if subsets of the tools are themselves compatible with one another, there is the risk of incompatibility among the tool subsets. To be truly effective, collections of tools must be compatible. Primarily this means it must be possible (and easy) to pass information among the tools. It also means the tools must all support a unifying conceptual model of software -- for example, they could all support a model of software as a collection of asynchronous, communicating processes.

With respect to these concepts, the STARS Program's objective to identify/develop assistance capitalizing on automation requires meeting the following more specific objective:

> develop and/or identify a compatible collection of manual and automated tools assisting a broad spectrum of personnel in performing a wide variety of the activities occurring during software creation and evolution.

5.7.2. Focus on Methodologies

Any collection of tools will reflect some general software creation and evolution approach. Frequently, some or all of the tools might be supportive of a variety of approaches. In either case, the general approach will, in effect, provide principles, practices and procedures guiding use of the tools, singly or in tandem. These general approaches are typically called "methodologies."

One characteristic of a methodology is its breadth of coverage of activities occurring during the life span of a software system. Ideally, a methodology will cover all of the activities. But most current methodologies cover only a (relatively small) portion of the life span, typically including, and frequently limited to, the coding portion.

Another methodology characteristic is its scope in terms of the variety of personnel supported by the methodology. Because of the generally disparate needs of the various personnel, it can be expected that most methodologies will be limited in scope to only one sector of the software creation and evolution personnel. For example, a technical methodology will service the needs of the technical personnel preparing the software requirement definitions, designs and code; and a management methodology will service the needs of project managers. A single project will typically require the services of many types of personnel. The project will therefore typically require multiple methodologies of varying scope, and compatibility among the methodologies will be an issue.

Methodologies and collections of tools are duals of each other. A methodology will imply a set of tools needed to support it. And a set of tools will support some collection of

methodologies. It is insufficient, therefore, for the STARS Program to attend merely to the development of collections of compatible tools. In addition, the Program must also address the closely related methodological issues. This leads to a second objective relating to the intent to identify/develop assistance capitalizing on automation:

> develop and/or identify a variety of methodologies that are dual to the tools and which, collectively or in sub sets, provide broad coverage and broad scope for defense systems software projects.

5.7.3. Integration Framework

Tool compatibility is primarily a characteristic visible to the tools' users. A methodology sets the stage for tool compatibility by establishing basic requirements for what it means for tools to be used together. With respect to the methodology, the tools either can or cannot be used together under the guidelines established by the methodology. If they cannot be used together, then this is (usually blindingly) apparent to the tools' users.

Another characteristic pertinent to tools is integration. Compatibility has to do with how well a tool "fits" with other tools in the eyes of the users. For example, the tools in a typical kitchen have evolved over time to provide a compatible collection. Although very similar to "compatibility" in meaning, "integratability" means the extent to which a tool adheres to a set of conventions for actualizing the tool. For example, the electronic tools in a modern kitchen all adhere to the standards which have evolved for plugging into electricity delivering sockets.

An integrated collection of tools, therefore, is one in which all of the tools adhere to a common set of conventions. For automated tools, these conventions can concern, among other things, the ways in which data is entered into or retrieved from a database management system, the representation used at the interface to the database management system, or the mechanisms used for invoking the tools. These conventions provide an integrating framework which, when implemented, offers a substrate upon which new tools can be implemented and thereby added to the existing collection of tools.

If different collections of tools employ the same integrating framework, then it will be easier to move tools among the collections. And, if a commonly accepted integrating framework evolves over time, then the task of creating new tools will be simplified since the developers of the new tools can target them for this framework with the knowledge that the result can be integrated into a wide variety of tool collections. This, in turn, may lead to the emergence of core collections of tools

having broad applicability and usable as a starting point for collections specific to particular applications.

Because of these potential benefits, the STARS Program has the following objective pertaining to the provision of automated assistance:

> provide an integrating framework enhancing the ability to transport tools and leading to core tool collections useful throughout the defense systems software community.

5.7.4. Transportability of Information

It is not reasonable to expect that a single, universally-used tool collection will evolve other time. If such were to happen, then it would mean everyone would be performing software creation and evolution in essentially the same way. Personal and organizational differences make this an impossibility. And preservation of a healthy competition in the acquisition of software makes it undesirable to force the evolution through legislation or policy.

The STARS Program must, therefore, prepare for the situation in which work on a single software system is supported by different tool collections over time. In the extreme, it must prepare for accepting delivery of software, and subsequently evolving it, without the ability to obtain any of the tools supporting the software's creation.

This means that, in addition to striving for a collection of compatible, integrated tools supporting Government-based or -supported software creation and evolution, the STARS Program should also prepare and promulgate a well-defined interface to this tool collection. Only with such as interface will it be possible to use the tool collection to support work on software developed using a different tool collection.

These considerations lead to the following objective for achieving the overall objective of providing automated assistance for software creation and evolution:

> define interfaces between the methodologies and other approaches to software creation and evolution, assuring the ability to effectively "import" a software system and work on it using the supporting tool collections.

5.7.5. Usability

The previous objectives decompose the intent to identify/develop automated assistance but do not directly address the attractiveness of the assistance. The primary concern with regard to attractiveness is the interface of the users to the tools. We consider this interface in this subsection.

Human engineering is the term usually applied to issues concerning the interface of users to an artifact. Here, we take it to apply to the ways in which the users interact with the tools, manual as well as automated.

One dimension of human engineering pertains to automated system concerns such as data display, data entry, and the system's command language. Another dimension pertains to ergonomic concerns such as the shape of the keyboard and the size of work station screens. Yet another dimension concerns the issue of how closely a newly introduced methodology and set of supporting tools mimics existing practice. We intend the human engineering of the system to include all these factors affecting its usability.

The necessity of assuring the automated assistance is attractive leads to the following objective:

> assure that the tools are well engineered for human use both individually and in subsets supporting one or more methodologies,

5.7.6. Progress and Value Assessment

Our final detailed objective comes from the already noted necessity of assuring that the technology actually leads to the intended improvement in product and process attributes, in particular, productivity, reusability, reliability and adaptability.

Two aspects of this assurance must be addressed. First, we must be sure that a relative improvement is being achieved -- this in essence means we have to be able to assure progress is being made. Second, we have to assure achieving some target level -- this in essence means we have to be able to make more absolute judgements of value.

These considerations lead to the final STARS Program objective:

> provide the means for measuring the effectiveness of the methodologies, the quality of the products produced by using them, and the benefit of the individual tools supporting them.

6. Scope and Goals of the STARS Activity Areas

The STARS Program is operationally divided into six areas: Methodology, Business Practices, Software Engineering Environments, Human Resources, Applications and Measurement. The focus of each area cuts across the objectives of the overall STARS Program as these are detailed above. This recognizes the need to group planning and execution activities on

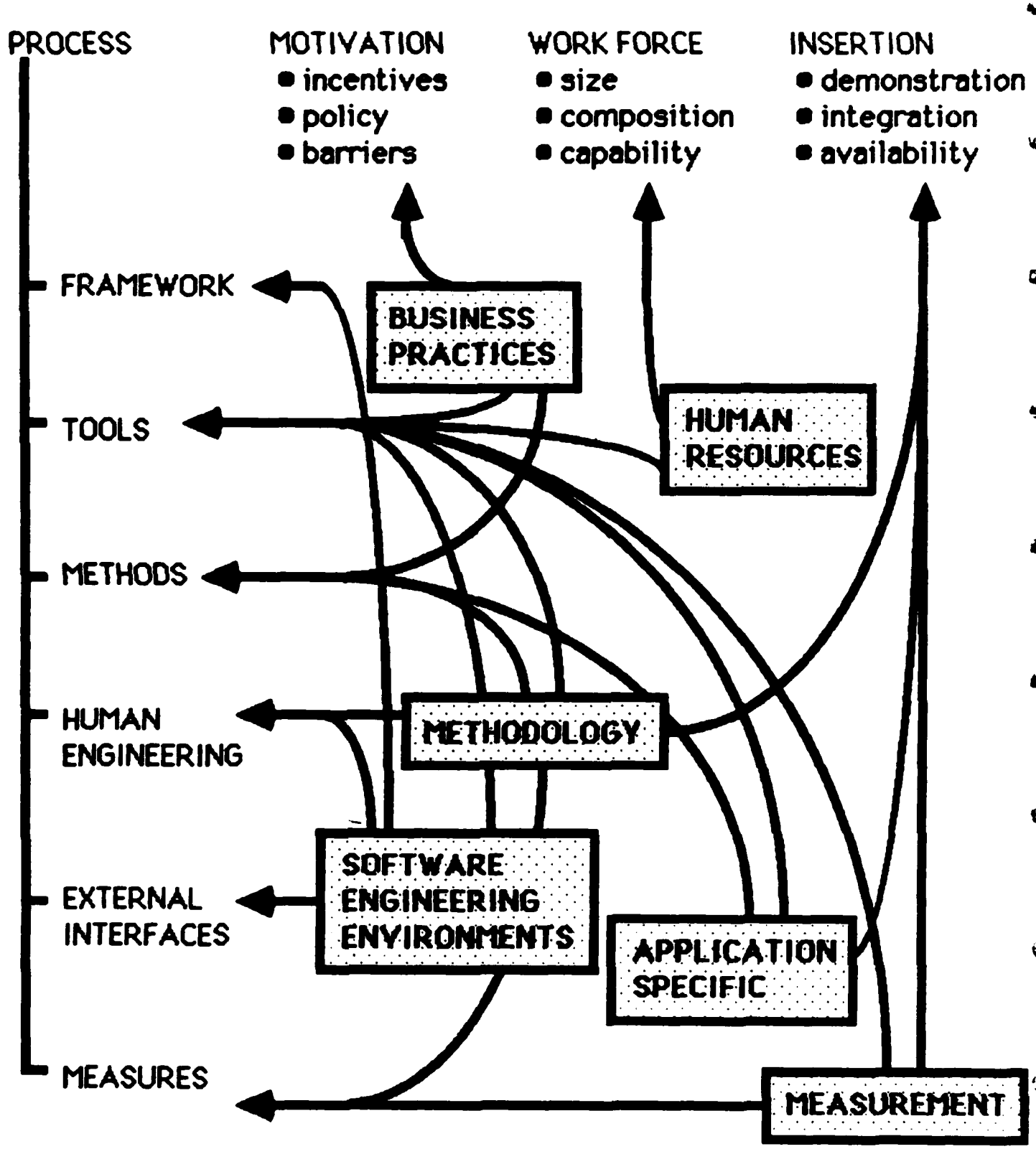


Figure 3: STARS Activity Areas

the basis of the underlying technology rather than on the basis of meeting a specific objective.

The Program's objectives do, however, establish the general context for the Activity Areas. Therefore, they can be used to specify the scope and major goals for each Activity Area. The scope of the Areas is graphically presented in Figure 3; it and the Areas' goals are discussed in this section.

6.1. General Responsibility for Tool Identification and Development

None of the Activity Areas has the total responsibility for identifying/developing tools. Rather, this responsibility is distributed across the areas, with each having responsibility for tools falling within its scope. We do not, therefore, mention specific types of tools in the following discussion of the individual areas. Rather, mentioning that a particular activity, like acquisition management, is within the scope of a particular area is meant to imply that area has responsibility for tools supporting the activity.

Because uncoordinated tool identification/development could negatively impact compatibility and integration, responsibility to assure these characteristics is centralized, within the Software Engineering Environments Area. This is discussed in the section on the Software Engineering Environments Area.

As a result of the duality between methodologies and tools and the fact that responsibility for tool development is distributed, the responsibility for identification/development of methodologies is also distributed. But, again, responsibility for coherency in the STARS methodology-related work is also centralized. This is discussed in the section on the Methodology Area, the area having the central responsibility in this case.

6.2. Methodology Area

The scope of the STARS Methodology Area is technical methodologies. With respect to these types of methodologies, the Methodology Area is responsible for identifying/developing methodologies valuable for defense systems software, comparatively assessing alternative methodologies, and demonstrating their value. In addition, the Methodology Area is responsible for raising the general level of consciousness of the DoD community with respect to methodological issues.

The Methodology Area is responsible for assuring the methodologies identified/developed by the STARS Program are broad in coverage (of activities supported) and in scope (of personnel serviced). To the extent that methodologies of different scope

result from the efforts of separate areas, the Methodology Area has responsibility to assure these methodologies are compatible.

The specific goals of the Methodology Area are:

develop and/or identify a variety of technical methodologies which, collectively or in subsets, provide broad coverage and broad scope for the technical aspects of defense systems software projects,

develop and/or identify tools supporting the technical methodologies,

develop and/or identify the means by which choices may be made among alternative methodologies,

assure the totality of methodologies resulting from the STARS Program are compatible and broad in coverage and scope.

6.3. Business Practices Area

The Business Practices Area is charged with assuring the practices of the DoD as a purchaser of software motivate the use of methodologies and tools resulting from the STARS Program. This includes issues of rights and data, acquisition policy, program management policy and acquisition/management regulations. It also includes the development of incentives that are not necessarily formulated in terms of policies and regulations.

Because of the close association between these topics and management/acquisition methodologies, this area has responsibility for these methodologies. It has the same responsibilities with respect to these methodologies that the Methodology Area has with respect to technical methodologies. It should advise the Methodology Area with respect to methodology coverage, scope, and compatibility issues from the standpoint of management/acquisition methodologies.

The Business Practices Area's specific goals are:

refine existing, Government "purchaser" incentives and policies, and define new ones, so "suppliers" of Government acquired software are motivated to use the automated assistance resulting from the STARS Program,

develop and/or identify a variety of methodologies which, collectively or in subsets, provide broad coverage and broad scope for the project management and acquisition aspects of defense systems software projects,

develop and/or identify tools supporting these management/acquisition methodologies.

6.4. Applications Area

The Applications Specific Area provides a link from the STARS Program to the community receiving the Program's results. It assures the methodologies and tools resulting from the Program are adequate and proper for a full spectrum of defense systems software. It also assures the receiving community is aware of the applicability of STARS-produced technology in specific application areas.

Because of the pertinence of reusability at the applications level, this area has the major responsibility for identifying/developing the technology of reusability. Code libraries are simple examples of such technology; it is expected the technology identified/developed by this area will address reusability issues throughout the total life span of software. The Applications Area has the responsibility to assure that methodologies and tools resulting from the STARS Program incorporate this reusability technology.

As the link to the using community, the Applications Area promotes the STARS identified/developed technology within the community. Primary in this regard is this area's responsibility to coordinate demonstrations of effectiveness for the methodologies and tools resulting from the STARS Program.

The specific goals of the Applications Area are:

develop and/or identify tools supporting activities occurring in specific application areas,

develop and/or identify tools supporting reusability,

foster use of the automated assistance by demonstrating its applicability to a wide variety of defense systems software projects.

6.5. Software Engineering Environments Area

Software engineering environments are collections of tools supporting modern software creation and evolution methodologies. The Software Engineering Environments Area assures the existence of the integrating framework(s) needed to host these tool collections for a wide spectrum of typical defense systems software projects. This integrating framework establishes a basic architecture for the software engineering environments produced as a result of the STARS Program and the Software Engineering Environments Area therefore has the primary responsibility for establishing and evolving this architecture. The framework also serves as an interface to the hardware and operating systems on which environments are hosted and the Software Engineering Environments Area therefore has the responsibility for the issues arising from these hosting concerns. In particular, it is responsible for

assuring that the results of the STARS Program are compatible with results from the CAIS effort and the various APSE implementation efforts, and for addressing issues regarding distributed, networked hosts.

The Software Engineering Environments Area has the responsibility to assure that information and tools can be transferred between software engineering environments. It oversees the development of methodology and tool collection interfaces allowing this transfer. The methodology interfaces should, in particular, allow the delivery of acquired software without the necessity to also deliver the tools used to develop the software. The tool collection interfaces should permit the transfer of tools between software engineering environments.

Coherency of the software engineering environments resulting from the STARS Program are also the responsibility of this area. The area's efforts should produce the generic, multiple-purpose tools supporting a variety of activities or personnel. And they should assure the tools produced through the efforts of other areas are compatible with these generic tools and with each other.

The area's specific goals are:

develop and/or identify generic tools useful across a broad range of software creation and evolution activities and supportive of the needs of a broad spectrum of personnel involved in software creation and evolution,

provide an integrating framework enhancing the ability to transport tools and leads to core tool collections useful throughout the defense systems software community,

define interfaces between the methodologies developed/identified through the STARS Program's efforts and other approaches to software creation and evolution, assuring the ability to "import" a software system and effectively work on it using the tool collections resulting from the STARS Program's efforts.

6.6. Human Resources Area

The Human Resources Area assures that sets of tools are individually and collectively attractive and that the work force is capable of effectively utilizing them. This includes the human engineering issues of the tools themselves and the collections of tools forming software engineering environments. It also includes education concerns that serve to modernize the capabilities of the work force so the tools and environments can be effectively used in the creation and evolution of defense systems software.

The Human Resources Area is also responsible for assuring the work force is of sufficient size and composition to meet the needs for defense system software. In this regard, the area is responsible for increasing the number of education programs preparing personnel for the work force, making it attractive for newly-trained personnel to enter the segment of the work force addressing defense systems software, and addressing problems of turn-over found in this segment of the work force.

The specific goals of this area are:

improve the size and composition of the work force and the general ability to use the automated assistance resulting from the STARS Program,

develop and/or identify tools assisting in the human engineering of end-applications systems,

assure all of the tools resulting from the STARS Program's efforts are well engineered for human use both individually and in subsets supporting one or more methodologies.

develop and/or identify educational technology.

6.7. Measurement Area

The Measurement Area assures that demonstrations of effectiveness and benefit can be done on a sound scientific basis. It should develop the measures and experimental approaches assisting in effective, cogent demonstrations. It should develop these measures and experimental approaches in response to the needs of the other areas within the STARS Program. It should serve as a consultant to the other areas in their demonstration efforts.

The Measurement Area also has the responsibility for demonstrating improvement over the current situation. It should establish exactly what the current baseline is and keep this knowledge up to date as improvements are made over time. It should assure demonstrations showing improvement (as opposed to demonstrations showing value) are made. In particular, it should assure that improvement in the productivity, reusability, reliability and adaptability are demonstrated.

The specific goals of the Measurement Area are:

develop and/or identify tools supporting the instrumentation and the analysis of data collected through this instrumentation (this instrumentation should support the evaluation of tools, collections of tools, and the products produced using the tools),

provide the means for measuring the value of the results of the STARS Program, in particular, the effectiveness of the methodologies, the quality of the products produced by using them, and the benefit of the individual tools supporting them,

foster use of the automated assistance by assuring demonstrations of effectiveness are made for the technology resulting from the STARS Program,

foster use of the automated assistance by assuring that improvement is demonstrated.

7. Management Concerns

The Program's goals and objectives and the derivative goals for the Activity Areas need to be further refined to reflect the production of intermediate results. These intermediate results are needed for management visibility and control. In this section, we present a general view of the phases in the lifetime of the STARS Program that can assist in this refinement.

The phases are presented in Figure 4. They are defined in terms of the "state of the world" achieved at three rough time points: late-1980's, early-1990's and mid-1990's.

By the late 1980's, the STARS Program should produce software engineering environments encompassing leading-edge technology and provide a perhaps not fully complete base (integrating framework and methodology interfaces) for using this technology.

The primary task in the early years of the STARS Program is to gain the knowledge of how to best utilize existing technology for the creation and evolution of defense software systems software. This is basically a task of consolidating existing technology, but it also involves assessing needs and preparing the work force so it is interested and willing to use modern, automation-based technology.

By the early 1990's, the software engineering environments should be much more complete and there should be multiple instances, tuned to various application areas. In addition, the environments should be readily judged as attractive and should be extensible so they can be readily kept up to date. It is unrealistic to assume these environments will encompass all and only the best of technology since there is a natural lag in packaging state-of-the-art technology so it can be transferred into wide-spread use. However, the technology encompassed by these early 1990's environments should not be farther than about three years behind advanced, state-of- the-art technology.

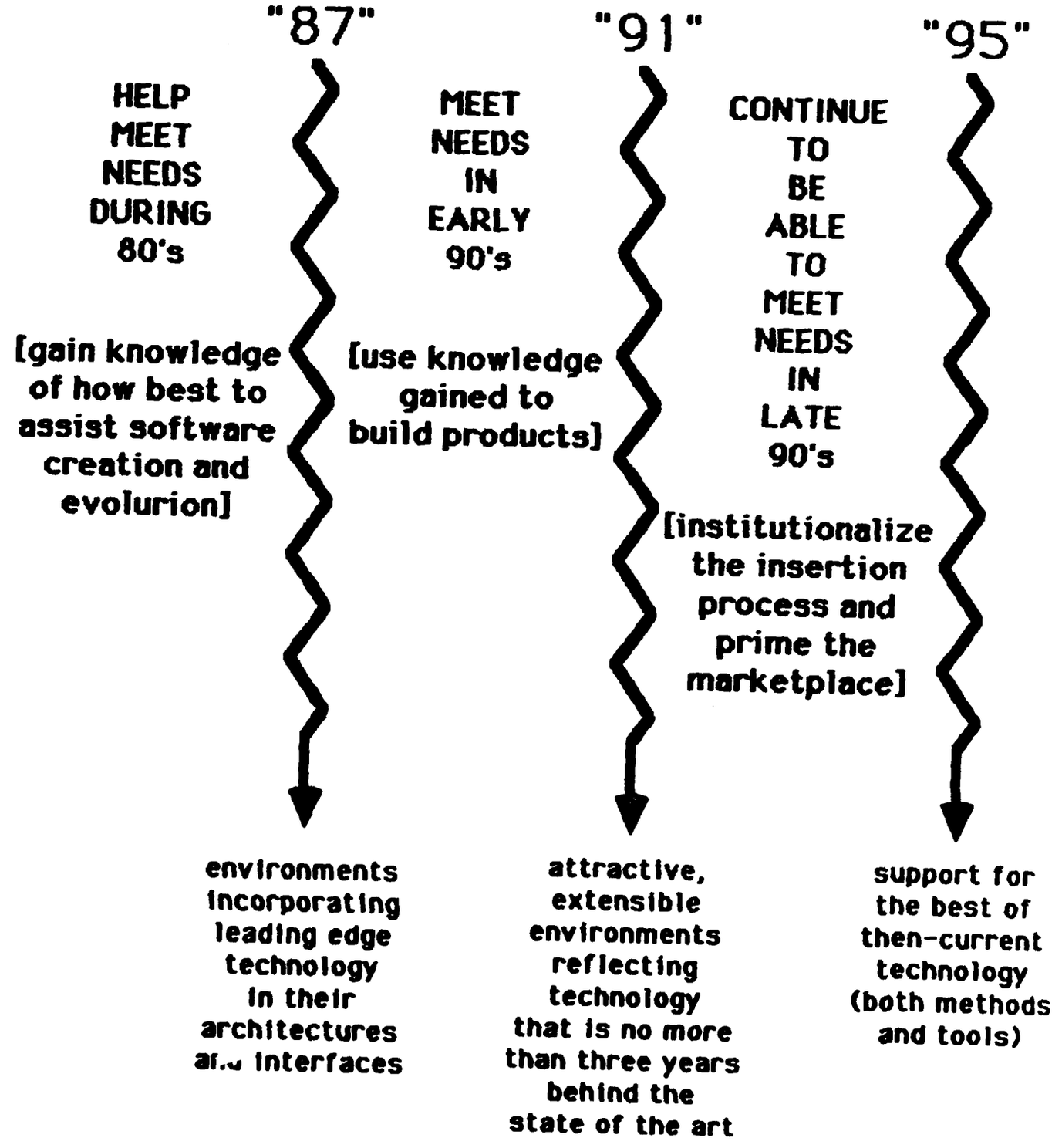


Figure 4: Phasing of the STARS Program

The task during the STARS Program's second phase is to capitalize on the knowledge of needs, benefits and possibilities acquired during the first phase and produce the software engineering environments discussed above. This is, therefore, essentially, a product building phase, where the products are final ones rather than the prototypes produced during the initial phase. (Activities oriented towards this phase's end goal will actually have to begin during the Program's first phase. During the first phase, however, they will be secondary and become the major thrust during the second phase.)

By the mid-1990's, the environments resulting from the STARS Program should be able to support the best of then-available technology. It is expected software creation and evolution technology will significantly mature by the mid-1990's, perhaps leading to approaches radically different from those used today. The environments stemming from the STARS Program's efforts in this time period should be able to incorporate this technology.

The task during the third phase is therefore radically different. Whereas the earlier phases have the intent to produce software engineering environments, prototypes at first and subsequently production versions, the task in the third phase is institutionalization. This means attention should turn to firmly establishing the policies and regulations fostering and assisting continual upgrading of the previously produced products so they are kept up to date and continue to be used for defense systems software creation and evolution. (As before, activities contributing to the aim of this third phase will be begun earlier, during the first and second phases. Also as before, they will be secondary during the earlier phases and become the major thrust during the third phase.)

8. Miscellaneous Concerns

8.1. Software Factory

The concept of a Software Factory came up several times during the meetings of the Goals and Objectives Working Group. It was not used as a mechanism for stating goals and objectives since it was felt to focus too much on "means" as opposed to "ends." However, it was felt to be of general utility for explaining the Program to various audiences, so the Group's observations about it are included here.

The concept of a Software Factory is broader than that of a software engineering environment. It is meant to encompass the variety of manual tools and practices which must be joined together with an automated environment to make it usable and effective. The analogy capitalizes on the fact that modern-day factories rely on automation to achieve time and cost effectiveness but encompass much more than automation.

The concept of a Software Factory immediately brings to mind the technical processes leading to the production of the Software Factory's products. Most obvious are the "manufacturing" processes which, for software, result in the executable code. Also included, however, are pre-implementation processes such as design and requirements definition. And, in addition to the technical processes, there are a variety of others, among them: educational (training the workers), management, acquisition, instrumentation, factory construction and expansion, and application specific. There are also variants of general technical processes which attend to special concerns, such as reusability.

There are several stages in the development of Software Factories*. These stages reflect a basic definition-design-implementation cycle:

determination of parts common between factories and common between facilities within a single factory

definition of requirements especially with respect to the specific products to be produced by specific factories

definition of specific factories

prototyping, demonstration and testing of these defined factories

validation and verification of the prototype factories

implementation of production-quality versions

system integration and operational test and evaluation

marketing, delivery, installation and support of the factories

Each Software Factory should be powerful (offering a broad range of modern technology), malleable (capable of being upgraded as new technology is discovered), attractive (of interest to the members of the work force as a place to work), and modern (always encompassing as near to the best technology as can be made effectively usable). More specific characteristics of the "ideal" Software Factory are: integrated, extensible, supportive of reusability, portable, flexible, tailorable, well engineered for human use, demonstrably of value, tuned to DoD management and acquisition needs and practices, usable to evolve products

---

* These stages also apply to software engineering environments.

created in other factories, and "self-learning." In addition, there should be a high degree of commonality among the factories so tools and personnel can be easily transferred.

8.2. Quantification

The Program's objectives, and as a result the Activity Area's goals, would be even more concrete if they could be quantified. This is not possible, however. We lack the means at this point to make quantitative assessment of software product and process attributes, in particular the ones of primary interest in the STARS Program. We cannot, therefore, either quantitatively assess the current situation or establish quantitative targets for the near or long term efforts of the STARS Program.

The means to quantify the current situation and measure progress are to be products of the Measurement Area. The Measurement Area could be directed to use the need to quantify the Program's objectives and the Areas' goals as a driver in their near-term efforts.

8.3. Products

At the moment, the emphasis in defining the STARS Program's products is upon software engineering environments. While this is without doubt an important area, this approach to defining the Program's products tends to emphasize only one segment of the overall Program.

One way in which to broaden this focus would be to talk about the Program's products in terms of a Software Factory, discussed above. This would bring the methodological and educational results of the Program into direct, rather than subsidiary, focus.

Another alternative would be to define the Program's products in terms of a "Software Engineering Handbook" for the DoD. This would shift the emphasis to technology in general rather than the automated tools that assist in using the technology. This could lead to easier definition of interfaces between the Areas and between the STARS Program itself and other Programs. Finally, it could make all of the Areas play an essentially co-equal role and lead to clearer definition of the major responsibilities of each Area.

8.4. Reorganization of the STARS Program

Under the current organization of the STARS Program, several topics are given global attention. While this is not necessarily bad, it does lead to some confusion as to lines of responsibility. It also leads to the need to assign responsibility for assuring the results coming from different areas are compatible.

In developing the Areas' goal statement given above, the Group noted the following topics are distributed across areas:

demonstrations of effectiveness and value, technology insertion in general,

product engineering for the full software engineering environments resulting from the STARS Program,

methodologies supporting defense systems software creation and evolution.

It would possibly be beneficial to redefine the organization of the STARS Program to reduce the extent to which these topics are distributed across Activity Areas. One approach would be to organize the Program along lines suggested by Figure 3. This would mean there would be areas responsible for insertion, work force concerns, motivation development and process definition and support. The breadth of responsibilities in the process definition and support area would warrant decomposing subareas for framework, tools, methods, human engineering, external interfaces and measurement.

Particularly problematic in the current organization is the somewhat uneven treatment given to the attributes of productivity, reusability, reliability and adaptability singled out for special attention in the mission statement and goals of the STARS Program. All of these attributes are made global concerns. But, none of the Activity Areas' goals make specific mention of productivity, reliability or adaptability, with the implication that all areas are to consider these attributes. Reusability is explicitly mentioned, in particular in the goals of the Application Area. But treatment of it is distributed with Applications developing/identifying the underlying technology, Methodology developing/identifying technical methodologies supporting reusability, Measurement demonstrating improvement in reusing software, Human Resources considering the human engineering aspects, and Business Practices developing/identifying management/acquisition methodologies supporting reusability and defining the incentives and policies fostering reusability.

Under the alternative organization suggested above, the treatment of the specially targeted attributes would be a bit clearer. Productivity would fall within the scope of the work force area because of its direct pertinence to this subject. Reusability would fall within the scope of the methods subarea since grappling with this issue is fundamentally a methodological issue. The process attributes of reliability and adaptability would fall within the scope of the tools subarea since it is through this group's effort that tools for general or specific purposes are developed/identified.

8.5. Underlying Problems

In its first meeting, the Goals and Objectives Working Group made note of a number of problems with the current defense systems software creation and evolution situation. In general, these problems are reflected in the statements of scope, goals and objectives. But the details do not show up. They are included here in the hope they may be of value to the STARS Joint Program Office in developing planning and reporting documents.

One class of problems reflects the fact that the work force performing defense systems software creation and evolution is limitied:

the size of the work force is insufficient

the productivity of individuals, and the work force as a whole, is insufficient

individuals are not sufficiently transferable across a broad range of projects

the task is too labor intensive

A second class of problems reflects attributes of the products produced during defense systems software creation and evolution:

the products are not secure

the products are not distributed

the products are not user friendly

the products do not utilize artificial intelligence technology

in general, the products are of poor quality; in particular, they are not maintainable, reliable or adaptable

software is not interoperable (that is, it cannot easily be moved from one operational situation to another)

requirements definitions are not clearly understandable to designers; designs are not clearly understandable to implementors

the performance of the products is not sufficient

the products exhibit a low degree of survivability

The third class of problems relates to the creation and evolution process itself:

the process is not sufficiently automated

the process is relatively ad hoc; it needs to be more well-defined

there is a lack of decisions aids and tools in general

there are few (comparative or absolute) measures for evaluating existing processes

in general, there is little ability to manage the process; in particular, there is little ability to predict its characteristics (length, cost, etc.) for specific projects

there are too many alternative processes and no way to choose among them

there is insufficient top-level management of the software creation and evolution process as different from processes for other artifacts

The final class of problems noted by the Group concern the time and cost effectiveness of the process:

it is difficult to decide where to invest additional resources necessary to get a project "back on track"

it is difficult to determine and state the requirements for software systems

there is unnecessary duplication; it is hard to decide whether a new software effort must start from scratch or whether an existing system can be used as a starting point

the software creation and evolution process is too labor intensive

there is an overburdening amount of bureaucracy associated with defense systems software efforts

portions of the process are unnecessarily sequentialized

there is a lack of general models, abstract objects and definitions of relationships among these models and objects

8.6. Relationship to Other Programs

There are a variety of other programs, both in and outside the Government, that share some of the STARS goals. In this section, we briefly discuss the relationship of the STARS Program to these other programs.

The DoD Software Engineering Institute, at Carnegie-Mellon University, was created as part of the planning for the STARS Program. The need for a technology transfer program, as an adjunct to the STARS Program, was realized early in this planning. The Software Engineering Institute was designed to meet this need. Also as a result of this planning, a central focus for the transfer of technology was not planned into the activities or organization of the STARS Program. Thus the relationship of the Software Engineering Institute to the STARS Program is distributed across the Program. The Institute should keep abreast of all of the developments within the STARS Program and position itself to absorb the results, integrate them with technology developed elsewhere, and transfer the result into wide-spread use throughout the DoD community. It should also advise the STARS Program on its plans, helping to make the results of the STARS Program compatible and complementary to developments elsewhere. Finally, it should actively assist in defining and conducting demonstrations of improvement, effectiveness and value.

The World-wide Military Command and Control System (WWMCCS) Information System Program seeks to upgrade WWMCCS through the use of Ada-based technology. As part of its activities, it is defining and implementing an Ada-based software engineering environment. In essence, this is an applications-oriented program but many of its results will likely be usable over a variety of applications. Its interface to the STARS Program is primarily through the Applications and Software Engineering Environments Areas.

The Strategic Defense Initiative is also basically an applications program as far as its relationship to the STARS Program is concerned. It does not have a centralized activity concerning software creation and evolution tools, but many of its activities will undoubtedly result in such tools. Its interface to the STARS Program is essentially the same as for the WWMCCS Information System Program.

The Ada Program shares the STARS Program's concern for software creation and evolution processes but is focused primarily on the implementation phase and the use of the Ada programming language. The interface between the Ada and STARS Programs is primarily through the Software Engineering Environments and Measurement Areas and concerns: integration frameworks, environment evaluation, implementation tools, and Ada software engineering environments.

The Very High Speed Integrated Circuit Program seeks to develop environments for the development of hardware utilizing the technology of very high-speed integration. It has been found that methodologies supporting this hardware development are very similar to software creation and evolution methodologies. It has also been found that environments support-

ing this development are similar to software engineering environments. The interface to the STARS Program therefore spans all of the technology-related areas within STARS.

There are a variety of programs outside the Government sector directly relating to the STARS Program. The academia-based programs (such as the Toolpack project at University of Colorado and the Arcadia project at the University of California, Irvine) relate primarily to the Software Engineering Environments Area. The industry-based programs (such as the Microelectronics Computer Consortium and the Software Productivity Consortium) are broad-based programs cutting across almost the full spectrum of STARS Program concerns -- the interface with these programs should be very broad.

The interface to foreign software technology programs should be similarly broad because of their generally broad scope. This includes the Alvey Program in the United Kingdom, the Esprit Program run by the European Economic Community, and the Sigma Program in Japan.

Distribution List for P-1855 (FINAL)

Computer Software and Systems Personnel

Mr. Joseph Batz
CSS Directorate
1211 S. Fern St., C107
Arlington, VA 22202

Dr. John Solomond
Army Deputy
STARS Program Office
1211 S. Fern St., C107
Arlington, VA 22202

Major Allen Kopp
Air Force AJPO Deputy
1211 S. Fern St., C107
Arlington, VA 22202

Lt. Col. John Leary
Air Force STARS Deputy
1211 S. Fern St., C107
Arlington, VA 22202

Dr. Edward Lieblein (25 copies)
Director
CSS Directorate
1211 S. Fern St., C107
Arlington, VA 22202

Dr. Clarence Giese
STARS Joint Program Office
1211 S. Fern St., C107
Arlington, VA 22202

Ms. Carol Morgan
Navy STARS Deputy
1211 S. Fern St., C107
Arlington, VA 22202

Major Frank Ward
Military Assitant to the Director
CSS Directorate
1211 S. Fern St., C107
Arlington, VA 22202

Col. Joe Greene
Director
STARS Joint Program Office
1211 S. Fern St., C107
Arlington, VA 22202

Ms. Virginia Castor
Director
Ada Joint Program Office
1211 S. Fern St., C107
Arlington, VA 22202

Service Managers

Commander, Army Materiel Development and Readiness Command
ATTN: DRCDE-SB, J. Hess
5001 Eisenhower Avenue
Alexandria, VA 22333

Col. Ken Nidiffer
HQ AFSC/ALR
Bldg. 1535, Room 205
Andrews AFB, MD 20331

Cmdr. Scott Gordon
Space & Naval Warfare Command
Code 03Y
Washington, DC 20363

CSED Review Panel

Dr. Dan Alpert, Director
Center for Advanced Study
University of Illinois
912 W. Illinois Street
Urbana, Illinois 61801

Dr. Barry W. Boehm
TRW Defense Systems Group
MS 2-2304
One Space Park
Redondo Beach, CA 90278

Dr. Ruth Davis
The Pymatuning Group, Inc.
2000 N. 15th Street, Suite 707
Arlington, VA 22201

Dr. Larry E. Druffel
Rational Machines
1501 Salado Drive
Mountain View, CA 94043

Dr. C.E. Hutchinson, Dean
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755

Mr. Oliver Selfridge
45 Percy Road
Lexington, MA 02173

Other

Richard DeMillo
Georgia Institute of Technology
School of Information and
Computer Science
Atlanta, GA 30332

Mary Shaw
Chief Scientist
SEI
Shadyside Place
South Aiken Avenue
Pittsburgh, PA 15232

John Manley
Software EngineeringInst.
Shadyside Place
South Aiken Avenue
Pittsburgh, PA 15232

Dennis Turner
CENTACS
ATTN: AMSEL-TCS-ADA
Ft. Mnmouth, NJ 07703

John Machado
Naval Electronic Systems Command
ELEX 613
2221 Jefferson Davis Highway
Crystal Plaza 6, Room 684
Washington, DC 20363

Major Chuck Lillie
HQ AFCS/PLRT
Andrews AFB, MD 20334-5000

Bill Riddle
software design & analysis
P.O. Box 3521
Boulder, CO 80303

Defense Technical Info. Center (2 copies)
Cameron Station
Alexandria, VA 22314

DoD-IDA Management Office
1801 N. Beauregard St.
Alexandria, VA 22311

IDA

Gen. W.Y. Smith, HQ
Mr. Seymour Deitchman, HQ
Mr. Robin Pirie, HQ
Ms. Karen Weber, HQ
Dr. John F. Kramer, CSED
Dr. John Salasin, CSED
Dr. Robert I. Winner, CSED (2 copies)
Ms. Cathy McDonald, CSED (2 copies)
Dr. Richard Morton, CSED (2 copies)
Ms. Sarah H. Nash, CSED (20 copies)
Mr. Sam Redwine, CSED (5 copies)
Ms. Katydean Price, CSED (2 copies)
IDA Control & Dist. Vault (5 copies)

END

12-86

DTIC